Evaluation of the Breadth First Search Algorithm with oneAPI/SYCL™on Intel® FPGAs

PC2 User Day
20 October 2023, Friday
Paderborn, Germany

Kaan Olgu(3rd year PhD Student)
PhD Supervisors : Prof Jose Nunez-Yanez, Prof Simon Mcintosh-Smith

University of BRISTOL

University of BRISTOL Alumni and Friends

SYCL™

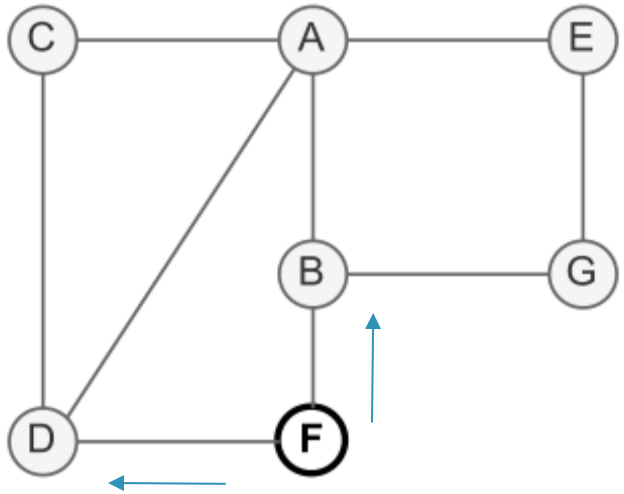
oneAPI

Breadth-First Search Algorithm

- **Definition**
  - A fundamental graph traversal algorithm.
  - Used for systematically exploring nodes in a graph or tree structure.
  - It starts at the root node and explores the neighbor nodes at the present depth prior to moving on to nodes at the next level.

    First level of neighbours : B, D
    Neighbours of neighbours (second iteration) (unvisited) : C, A, G
    3rd level : E

  - Guarantees shortest path for unweighted graphs.



- **Why Learn BFS?**
  - Essential for solving various real-world problems
  - Example : Route planning in GPS navigation systems, social media, P2P networks.

- **Performance Bottlenecks :**
  - Irregular memory accesses rather than computation intensity!
  - Dataset sizes (trillions of edges)
  - Next to visit is decided during the execution

Why Use oneAPI with Intel FPGAs

- **High Level Language**
  - OneAPI is based on SYCL which is an abstraction layer for C++ heterogeneous computing
- **Unified Programming Model**
  - Same code could be run for CPU,GPU and FPGA (with a bit of modification!)
- **Portability**
  - Abstracts away the underlying hardware architecture, making it easier to move applications between different types of devices..
- **Ease of Use**
  - Makes it easier for developers to transition to using OneAPI compared to learning specialized hardware description languages like VHDL.
- **Heterogeneous Computing**
  - Part of the code could be offloaded to GPU and some parts to FPGA in the same source file

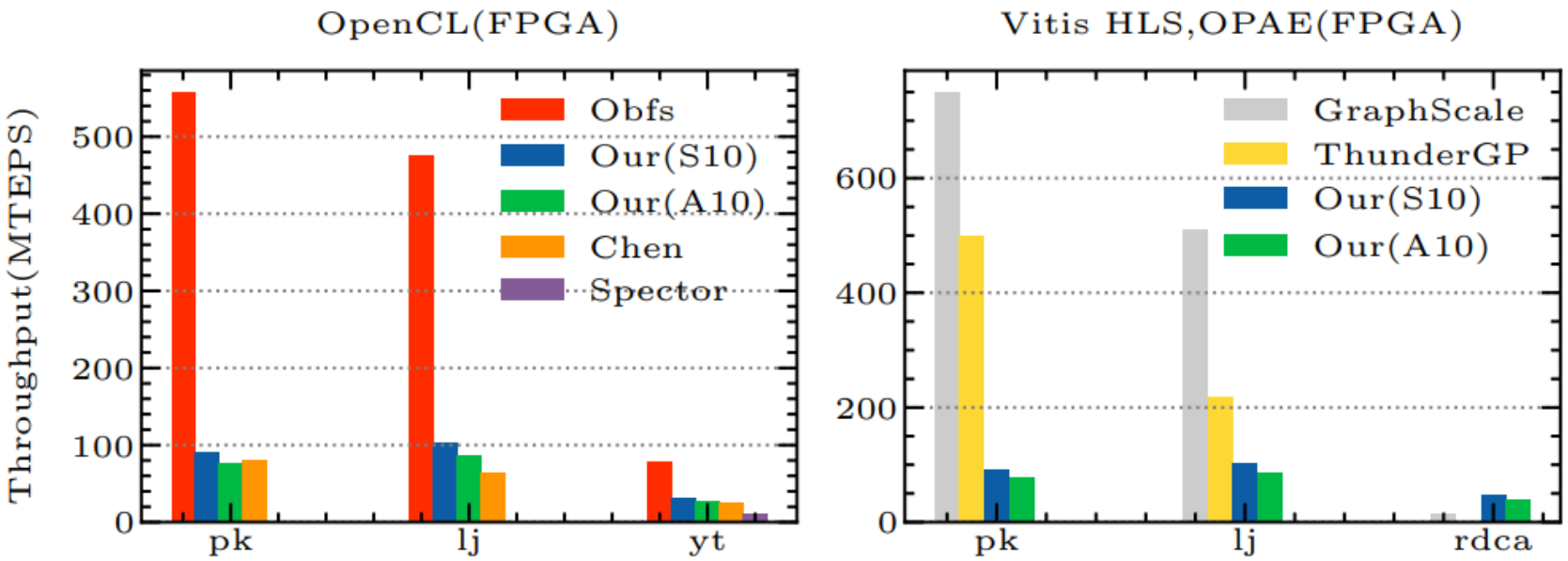
- **Why to use FPGAs as accelerators ?**
  - Reprogrammable
  - Lower power consumptions
  - Lower latencies

State of the Project

Overview of the project

- Research innovation and advancing performance of a relatively new programming language
- Target to achieve better performance with high level languages on FPGAs compared to more mature OpenCL and Vitis HLS
- With the help from Tobias Kenter, Heinrich Riebler, Michael Laß at the Hackathon, we managed to improve our performances 25% back in September at the Paderborn! Work in progress!

(OLD) Real-World Dataset Performance Comparison



Conclusion

- The source code is publicly accessible at https://github.com/kaanolgu/bfs-sycl-fpga
- Special thanks for the organisers and Paderborn Center for Parallel Computing!

Thank You!